



# Conception Web Avancé

## Rapport du projet

Maxime COLLIAT  
Rahman YILMAZ  
Yoan DUSOLEIL  
Rémy BARRANCO  
Julie PRIGENT  
Betul SENER

# **SOMMAIRE**

<b>I) Contexte</b>	<b>3</b>
<b>II) Cahier des charges</b>	<b>3</b>
<b>III) Diagramme</b>	<b>5</b>
<b>IV) Conception</b>	<b>8</b>
<b>V) Démonstration</b>	<b>12</b>
<b>VI) Conclusion</b>	<b>17</b>

## **I) Contexte**

L'objectif de ce projet vise à développer une application de gestion de tâches offrant aux utilisateurs la possibilité de créer, afficher, mettre à jour et supprimer leurs tâches à accomplir.

Les fonctionnalités du projet comprennent une page d'accueil qui affiche la liste des tâches, permettant aux utilisateurs de filtrer leurs tâches par statut et priorité, ainsi que de trier par date d'échéance ou priorité. Une page de création de tâches permet aux utilisateurs d'ajouter de nouvelles tâches. Une page de détails de tâche affiche les informations de chaque tâche et permet de les marquer comme terminées. Une page de modification de tâche permet de mettre à jour les détails existants et enfin, une page de suppression de tâche.

Les technologies qui seront à utiliser incluent Angular, Angular Router pour la navigation et nous avons la possibilité de créer un back-end pour la gestion du stockage des données.

Nous avons fait le parti-pris de ne pas utiliser de librairie de composants graphiques comme Material. En effet, le projet ayant pour objectif principal d'évaluer nos compétences en TypeScript avec le framework Angular, il nous paraissait plus pertinent de faire nos composants de zéro par nous-même.

## **II) Cahier des charges**

**Nom de l'Application :** Cadmium

**Objectif :** Développer une application Web de gestion des tâches en utilisant Angular et les technologies du Web.

**Fonctionnalités Requises :**

### **Page d'Accueil :**

- Affichage de la liste des tâches à faire.
- Possibilité de filtrer les tâches par statut (en cours, terminées) et par priorité (élevée, moyenne, faible).
- Capacité de trier les tâches par date d'échéance ou par priorité.

### **Page de Création de Tâches :**

- Permettre aux utilisateurs de créer de nouvelles tâches en spécifiant un titre, une date d'échéance et une priorité.
- Valider les entrées de l'utilisateur et afficher des messages d'erreur en cas de saisie incorrecte.

### **Page de Détails de Tâche :**

- Affichage des détails d'une tâche, y compris son nom, sa date d'échéance, sa priorité et son statut.
- Possibilité pour les utilisateurs de marquer la tâche comme terminée.

### **Page de Modification de Tâche :**

- Permettre aux utilisateurs de modifier les détails d'une tâche existante, y compris son titre, sa date d'échéance, sa priorité et son statut.
- Valider les entrées de l'utilisateur et afficher des messages d'erreur en cas de saisie incorrecte.

### **Page de Suppression de Tâche :**

- Permettre aux utilisateurs de supprimer une tâche existante après confirmation.

### **Exigences Techniques :**

L'application doit être développée en utilisant les technologies suivantes :

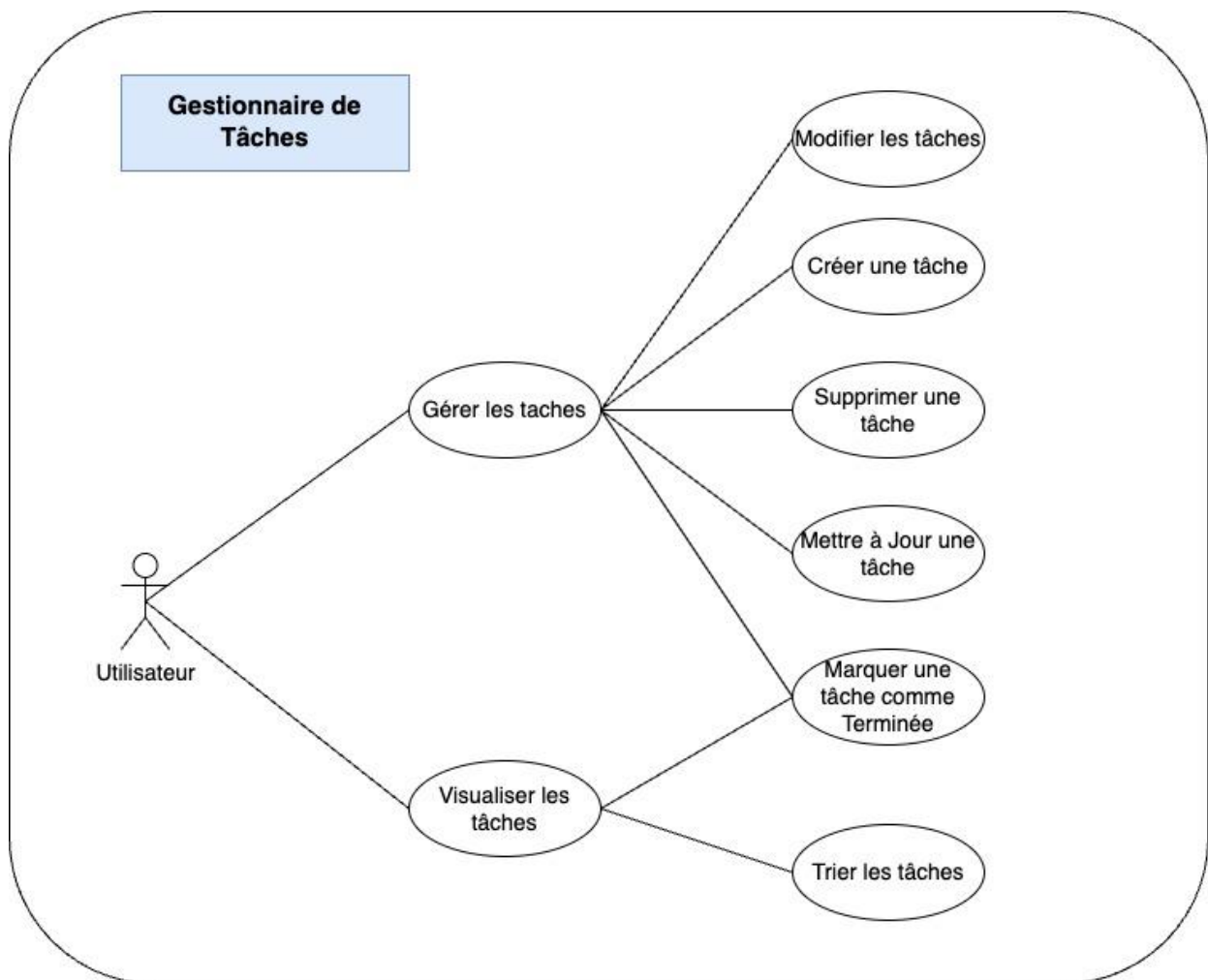
- Angular (avec Angular CLI) pour le développement de l'application front-end.
- Angular Router pour la gestion de la navigation entre les différentes pages.

## **III) Diagramme**

Voici quelques diagrammes que nous avons trouvé utile à partager pour une meilleure compréhension ce projet :

## Diagramme USE CASE :

Voici le schéma du diagramme Use Case de l'application :



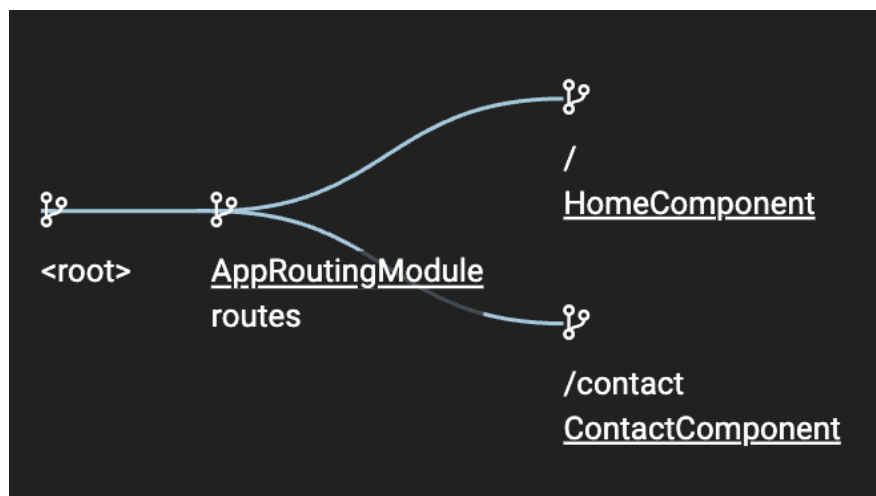
### Description :

1. L'utilisateur peut créer une nouvelle tâche en spécifiant son titre, sa date d'échéance et sa priorité.

2. L'utilisateur peut consulter la liste des tâches à faire, filtrer les tâches par statut et priorité et trier les tâches par date d'échéance ou priorité.
3. L'utilisateur peut mettre à jour les détails d'une tâche existante, y compris son titre, sa date d'échéance, sa priorité, et son statut.
4. L'utilisateur peut marquer une tâche comme terminée.
5. L'utilisateur peut supprimer une tâche existante après confirmation.

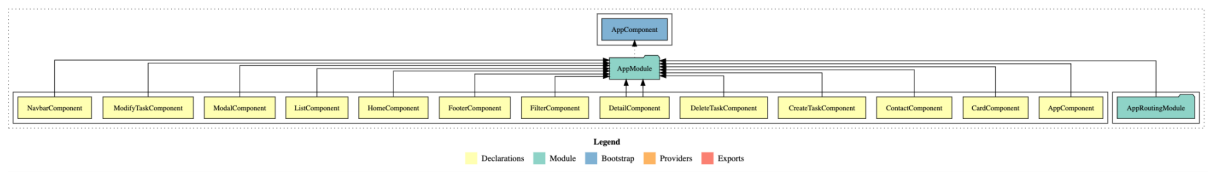
### Diagramme des Routes du projet :

Voici le schéma du diagramme de la classe avec les liens :



### Diagramme des Components du projet :

Voici le schéma du diagramme de la classe avec les liens :



Retrouvez tous les diagrammes dans leur qualité original dans le github du projet à ce lien :

<https://github.com/Phaired/Projet-CWA/tree/main/rapport/screenshot/diagramme>

## IV) Conception

La partie conception regroupe toutes les explications sur les différentes implémentations de ce projet.

### Tâche :

La classe "Tache.ts" permet de modéliser une tâche dans notre site. Une tâche est caractérisé par les attributs :

1. **id** (number) : Cet attribut représente l'identifiant unique de la tâche. Chaque tâche possède un identifiant qui la distingue de manière unique des autres tâches.
2. **intitule** (string) : Cet attribut correspond au nom ou à l'intitulé de la tâche. Il s'agit d'une courte description ou d'un titre permettant d'identifier la nature de la tâche.



3. **date\_creation** (Date) : Il s'agit de la date à laquelle la tâche a été créée. Cela peut être utile pour suivre la chronologie des tâches et pour établir un historique.
4. **date\_fin** (Date) : Cet attribut représente la date d'échéance ou la date prévue pour la réalisation de la tâche. Il indique quand la tâche devrait être terminée.
5. **description** (string) : La description fournit des détails plus complets sur la tâche. Cela peut inclure des informations supplémentaires sur ce qui doit être fait, des instructions spécifiques, etc.
6. **priority** (number) : La priorité de la tâche, souvent exprimée numériquement, indique son importance relative par rapport à d'autres tâches. Une valeur plus élevée peut indiquer une priorité plus grande.
7. **is\_terminate** (boolean) : Cet attribut indique si la tâche est terminée ou non. S'il est vrai, cela signifie que la tâche est complète ; sinon, elle est toujours en cours.
8. **color** (string) : La couleur associée à la tâche. Cela peut être utilisé à des fins de visualisation ou de catégorisation. Par exemple, des couleurs différentes peuvent être associées à des niveaux de priorité différents ou à des types de tâches spécifiques.

### **Service LocalStorageRepository :**

Ce service permet de gérer la sauvegarde et la suppression des tâches dans le projet. Il stocke la liste des tâches en tant qu'objet JSON dans le localStorage du navigateur pour offrir une persistance des données sans avoir à recourir à une base de données et une API.

### **Component Filter :**

Ce composant prend en paramètre une liste de tâches et offre des fonctions permettant de trier cette dernière selon :

- La date

- Le niveau de priorité

Il permet aussi de filtrer la liste selon :

- Le statut (Finie/Non finie)
- La priorité (Faible, Moyenne, Haute)

(Trucs restants à faire)

### **Modal :**

Le composant modal joue un rôle central dans l'application en simplifiant la création de fenêtres contextuelles. Il permet d'afficher facilement une popup stylisée en passant simplement le composant à afficher dans la modalité. Cette approche simplifiée réduit la complexité du code et offre une solution cohérente pour présenter des informations contextuelles.

### **Les component d'une Tâche :**

Une tâche est associée à différents composants qui permettent à l'utilisateur de créer, modifier ou supprimer une tâche. On retrouve les composants :

- **card** : Chaque tâche possède un composant visuel qui permet d'afficher ses informations globales dans l'accueil du site.
- **detail** : Ce composant permet d'afficher au sein d'un composant modal l'ensemble des paramètres d'un objet Task tel que son titre, sa description, sa priorité, sa date butoir et la couleur attribuée.
- **create-task** : Le composant de création de tâche permet à l'utilisateur de créer une tâche et l'ajouter dans la liste des tâches dans la page d'accueil. Il offre un formulaire interactif où l'utilisateur peut saisir les modifications nécessaires telles que le titre mis à jour, une description modifiée, ou d'autres champs pertinents.

- **modify-task** : Le composant de modification de tâche permet à l'utilisateur de mettre à jour les détails d'une tâche existante. Il offre un formulaire interactif où l'utilisateur peut saisir les modifications nécessaires telles que le titre mis à jour, une description modifiée, ou d'autres champs pertinents. Ce composant est similaire à celui de la création de tâche.
- **delete-task** : Le composant de suppression de tâche est responsable de la confirmation de la suppression d'une tâche. Lorsque l'utilisateur souhaite supprimer une tâche, le composant de suppression est activé pour recueillir une confirmation. Il affiche le titre de la tâche concernée et propose un bouton de suppression, permettant à l'utilisateur de confirmer son choix de manière explicite.

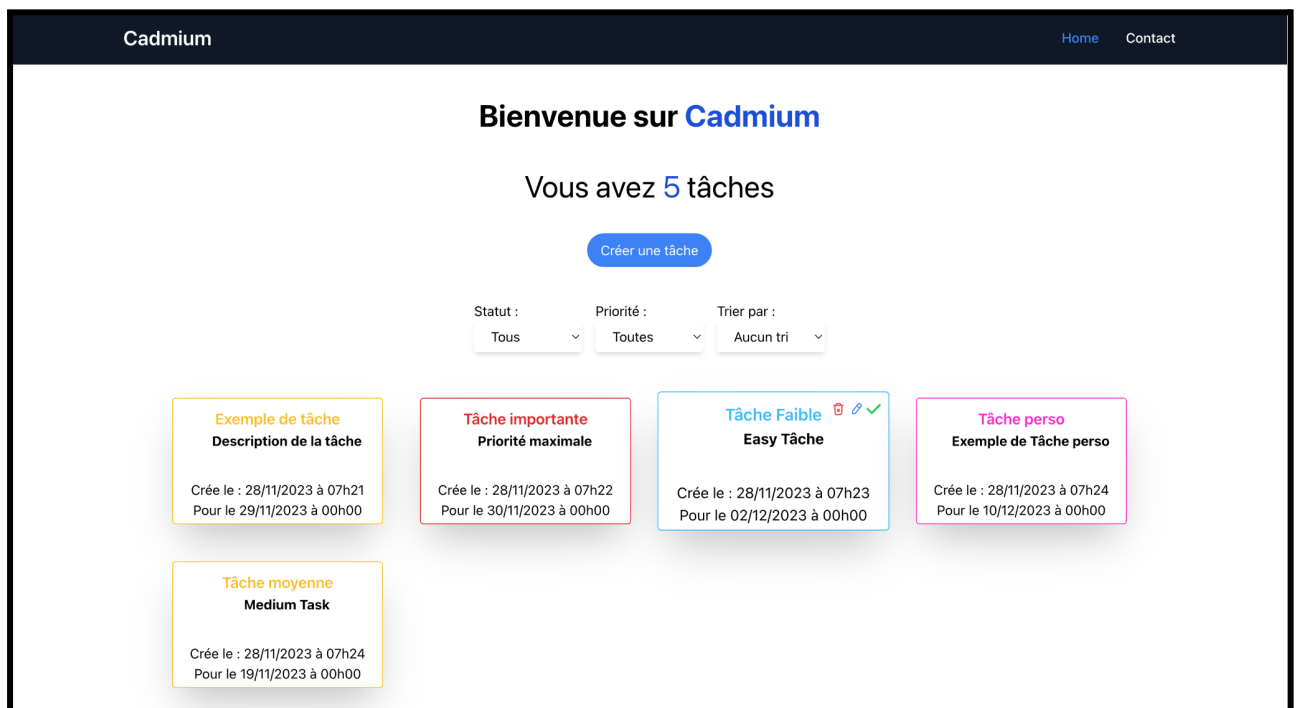
### **Devops :**

Afin de faciliter notre développement, nous avons choisi d'utiliser GitHub avec quelques spécificités. En effet, un pull request à partir d'une branche vers la branche principale est nécessaire pour modifier cette dernière. Cela nous permet d'éviter que la branche principale soit modifiée sans une revue. Nous avons également opté pour l'utilisation de Prettier, assurant un formatage uniforme du code. Pour garantir la cohérence, une GitHub Action vérifie le bon formatage au moment de la pull request, affichant une erreur en cas de mauvais formatage.

## V) Démonstration

Voici une démonstration rapide des différentes pages et fonctionnalités du projet.

### 1. Page d'Accueil :



## 2. Page de Création de Tâches :

×

### Ajouter une Tâche

Nom de la tâche

Faire ses devoirs

Le nom de la tâche doit contenir au moins 3 caractères

Rendre le projet CWA

Le nom de la tâche doit contenir au moins 10 caractères

Priorité : 1

Date de fin

jj/mm/aaaa

Choisissez une date

Couleur de la tâche

### 3. Page de Détails de Tâche :

## Tâche importante

Priorité maximale

Priorité : 3

Date de création: 28/11/2023

Date de fin: 30/11/2023

Couleur de la tâche

Etat de la tâche

☐

### 4. Page de Modification de Tâche :

×

## Modifier une Tâche

Nom de la tâche

Exemple de tâche

Description de la tâche

Priorité : 2

Date de fin

29/11/2023

Couleur de la tâche

Modifier la Tâche

## 5. Page de détail de la tâche (au clic d'une tâche sur la page d'accueil)



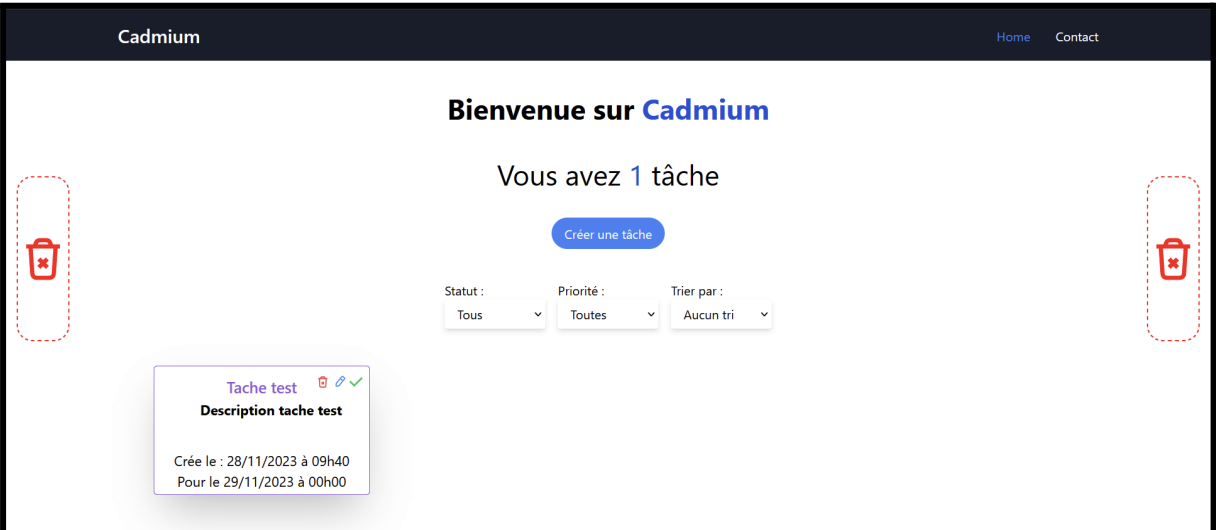
A modal window titled "Tache test" with a close button (X) in the top right corner. The content includes a dark grey bar with the text "Description tache test". Below this, there is a "Priorité : 1" label with a horizontal slider. The "Date de création:" is 28/11/2023 and the "Date de fin:" is 29/11/2023, both in dark grey boxes. A "Couleur de la tâche" section shows a purple square. The "Etat de la tâche" section has a small square checkbox.

## 6. Page de Suppression de Tâche :



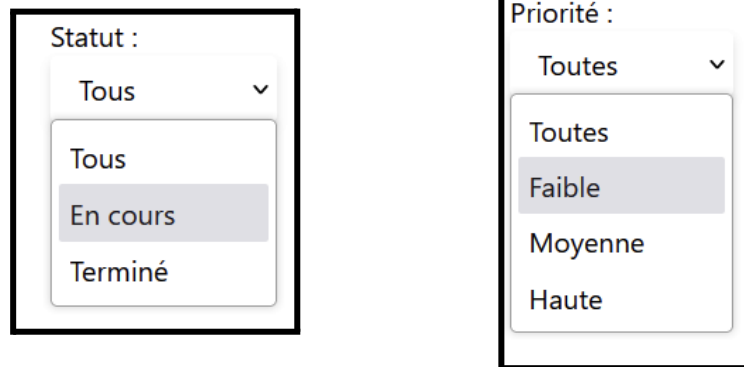
A modal window with a close button (X) in the top right corner. It contains two buttons: a red "Supprimer" button and a blue "Annuler" button.

## 7. Page de Suppression de tâche en glissé-déposé :

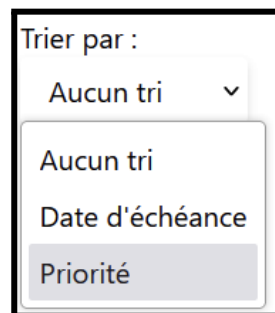


The main application interface for "Cadmium". The header has "Cadmium" on the left and "Home" and "Contact" on the right. The main content area says "Bienvenue sur Cadmium" and "Vous avez 1 tâche". Below this is a blue "Créer une tâche" button. There are three filter dropdowns: "Statut : Tous", "Priorité : Toutes", and "Trier par : Aucun tri". On the left and right sides, there are dashed red boxes containing a trash can icon. At the bottom left, there is a task card for "Tache test" with a description, creation date (28/11/2023 à 09h40), and due date (29/11/2023 à 00h00). The card has icons for delete, edit, and checkmark.

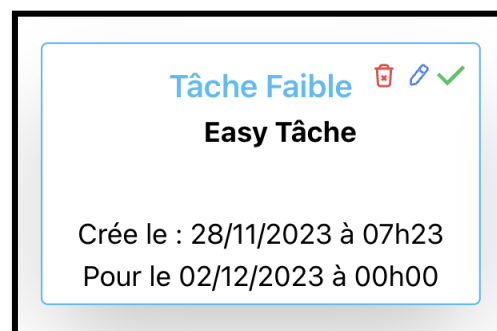
## 8. Détail des filtres sur la page d'accueil



## 9. Détail des critères de tri sur la page d'accueil



## 10. Action rapide sur une tâche

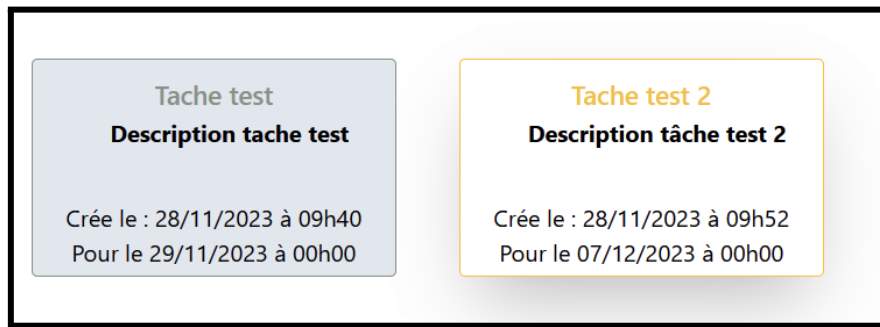


Les actions rapides sur une tâche sont les suivantes :

1. Poubelle : Supprimer la tâche pointée par le curseur.
2. Crayon : Modifier la tâche pointée par le curseur.
3. Check : Mettre la tâche en tant que terminé.

## 11. Affichage de la tâche grisée (suite au clic du logo coche vert pour signaler que la tâche est terminée)





## **VI) Conclusion**

### **1. Améliorations**

A l'issue de la réalisation de notre application de gestion des tâches nous souhaitons porter un regard critique sur notre travail. A cette fin nous détaillerons dans cette partie les évolutions envisageables de certaines des fonctionnalités de notre application.

- **Affichage du détail d'une tâche :**
  - Afin de rendre l'application plus ergonomique il serait intéressant d'ajouter un bouton permettant de modifier la tâche depuis ce visuel, en effet actuellement il faut retourner sur l'accueil et cliquer sur le logo correspondant.
  - Pour les mêmes raisons, nous pourrions modifier l'état de la tâche en cochant ou décochant la case qui indique l'état de la tâche depuis ce visuel également.
- **Création d'une tâche :**
  - Afin de rendre l'utilisation de l'application fluide nous pourrions mettre la date du jour en date par défaut lors de la création d'une

tâche, afin de ne pas être bloqué lorsque nous n'avons pas besoin de ce paramètre ou que nous souhaitons le définir ultérieurement.

- Pour rendre l'application adaptée à tous types de tâche, nous pourrions mettre en place un principe de tâche parent qui soit liée à des sous tâches.
- Ajout de la possibilité de dupliquer une tâche.

## **2. Conclusion**

Pour conclure de ce projet, nous avons répondu à tous les points du cahier des charges de ce projet. Ce projet nous a permis de nous familiariser avec le framework Angular et de maîtriser les différents éléments qui facilitent le temps de développement d'applications web modernes.

Vous pouvez retrouver le lien du repository Github du projet à ce lien : <https://github.com/Phaired/Projet-CWA>